SAHELI KAR

Project 5

Contents

Data:

I am using an order data which contains the record of order. I saved the data in csv format. The data in excel format is given below:

OrderDate	Region	Rep	Item	Units	Cost	Total
1/6/2012	East	Jones	Pencil	95	1.99	189.05
1/23/2012	Central	Kivell	Binder	50	19.99	999.5
2/9/2012	Central	Jardine	Pencil	36	4.99	179.64
2/26/2012	Central	Gill	Pen	27	19.99	539.73
3/15/2012	West	Sorvino	Pencil	56	2.99	167.44
4/1/2012	East	Jones	Binder	60	4.99	299.4
4/18/2012	Central	Andrews	Pencil	75	1.99	149.25
5/5/2012	Central	Jardine	Pencil	90	4.99	449.1
5/22/2012	West	Thompson	Pencil	32	1.99	63.68
6/8/2012	East	Jones	Binder	60	8.99	539.4
6/25/2012	Central	Morgan	Pencil	90	4.99	449.1
7/12/2012	East	Howard	Binder	29	1.99	57.71
7/29/2012	East	Parent	Binder	81	19.99	1,619.19
8/15/2012	East	Jones	Pencil	35	4.99	174.65
9/1/2012	Central	Smith	Desk	2	125	250
9/18/2012	East	Jones	Pen Set	16	15.99	255.84
10/5/2012	Central	Morgan	Binder	28	8.99	251.72
10/22/2012	East	Jones	Pen	64	8.99	575.36
11/8/2012	East	Parent	Pen	15	19.99	299.85
11/25/2012	Central	Kivell	Pen Set	96	4.99	479.04

12/12/2012	Central	Smith	Pencil	67	1.29	86.43
12/29/2012	East	Parent	Pen Set	74	15.99	1,183.26
1/15/2013	Central	Gill	Binder	46	8.99	413.54
2/1/2013	Central	Smith	Binder	87	15	1,305.00
2/18/2013	East	Jones	Binder	4	4.99	19.96
3/7/2013	West	Sorvino	Binder	7	19.99	139.93
3/24/2013	Central	Jardine	Pen Set	50	4.99	249.5
4/10/2013	Central	Andrews	Pencil	66	1.99	131.34
4/27/2013	East	Howard	Pen	96	4.99	479.04
5/14/2013	Central	Gill	Pencil	53	1.29	68.37
5/31/2013	Central	Gill	Binder	80	8.99	719.2
6/17/2013	Central	Kivell	Desk	5	125	625
7/4/2013	East	Jones	Pen Set	62	4.99	309.38
7/21/2013	Central	Morgan	Pen Set	55	12.49	686.95
8/7/2013	Central	Kivell	Pen Set	42	23.95	1,005.90
8/24/2013	West	Sorvino	Desk	3	275	825
9/10/2013	Central	Gill	Pencil	7	1.29	9.03
9/27/2013	West	Sorvino	Pen	76	1.99	151.24
10/14/2013	West	Thompson	Binder	57	19.99	1,139.43
10/31/2013	Central	Andrews	Pencil	14	1.29	18.06
11/17/2013	Central	Jardine	Binder	11	4.99	54.89
12/4/2013	Central	Jardine	Binder	94	19.99	1,879.06
12/21/2013	Central	Andrews	Binder	28	4.99	139.72

Data loading:

I loaded the csv file in R into a dataframe. I created separate dataframes for the nodes to be created in Neo4j and exported the dataframes into separate csv files. The R script for data loading cleaning up and saving into file is provided below:

```
1. require(dplyr)
2. orderdf<- read.csv(file.choose(), header=TRUE)
3.
4. Product<-unique(select(orderdf, Item))
5. write.csv(Product, file = "Product.csv")
6.
7. Rep<-unique(select(orderdf, Rep, Region))
8. write.csv(Rep, file = "Representative.csv")
```

Two types of nodes are created from the Product.csv file and Representative.csv file. They are - Product and Rep. The properties for the nodes are provided below:

Product – Item

Rep – Name, Region

The data load process with the node creation in Neo4j is provided below:

```
load csv with headers from "file:C:\\data\\Product.csv" as product create (p1:Product {Item:product.Item})

load csv with headers from "file:C:\\data\\Representative.csv" as rep create (n1:Rep {Name:rep.Rep, Region:rep.Region})
```

The Orders relationship id created between the Rep and Product. The properties for Orders relationship is provided below:

Orders: Date, Units, Cost, Total

The script to load the Orders relationship is provided below:

```
load csv with headers from "file:C:\\data\\Order_data.csv" as order match (n1:Rep {Name:order.Rep}), (p:Product{Item:order.Item}) create (n1)-[r:Orders{Date:order.OrderDate, Units:order.Units, Cost:order.Cost, Total:order.Total}]->(p)
```

Cypher query:

Show all the representatives who ordered Pencil along with the OrderDate and the Cost:

```
match(n1:Rep)-[r:Orders]->(p:Product{Item:"Pencil"}) return n1,r.Date,r.Cost
```

n1	r.Date	r.Cost
{"Name":"Jones","Region":"East"}	1/6/2012	1.99
{"Name":"Jones","Region":"East"}	8/15/2012	4.99
{"Name":"Jardine","Region":"Central"}	2/9/2012	4.99
{"Name":"Jardine","Region":"Central"}	5/5/2012	4.99
{"Name":"Gill","Region":"Central"}	5/14/2013	1.29
{"Name":"Gill","Region":"Central"}	9/10/2013	1.29
{"Name":"Sorvino","Region":"West"}	3/15/2012	2.99
{"Name":"Andrews","Region":"Central"}	4/18/2012	1.99
{"Name":"Andrews","Region":"Central"}	4/10/2013	1.99
{"Name":"Andrews","Region":"Central"}	10/31/2013	1.29

{"Name":"Thompson","Region":"West"}	5/22/2012	1.99
{"Name":"Morgan","Region":"Central"}	6/25/2012	4.99
{"Name":"Smith","Region":"Central"}	12/12/2012	1.29

Show all the representatives from East:

```
1. match(n1:Rep{Region:"East"}) return n1
```

n1
{"Name":"Jones","Region":"East"}
{"Name":"Howard","Region":"East"}
{"Name":"Parent","Region":"East"}

The advantages and disadvantages of having this information in a graph database instead of a relational database

Here in Neo4j we can logically map the entities and the relationships between them, while if we have to implement the same thing in Relational database we have to first normalize the data and split them into three fact tables – Representative, Product and Order table. The Representative table holds the ID of the representative, Name and Region. The order table will hold the Item Name and the ID. Order table will be the master table and it will hold all the information of the order data. Though there will be referential integrity between the tables but the representation of the data in Graph database will be more logical which actually shows the relationships between the entities.

The retrieval of data in Graph database will be faster than relational database. Suppose we have to order all the details of ordering a Pencil. We have to join two tables with the Order table and retrieve the values. We know joining is always slower. While in Graph database already stores the relationship and we don't have to create relationship in between the entities.

Graph database use more space than relational database. In my example the data is very small but if we use large datasets , it will utilize lot of memory space than Relational Database.